Network Working Group
Request for Comments #662
NIC #31387

Raj Kanodia
Project MAC, MIT
November 26, 1974

PERFORMANCE IMPROVEMENT IN ARPANET FILE TRANSFERS FROM MULTICS

With the growth of Multics usage in the ARPA Network community, users often need to transfer large amounts of data from Multics to other systems. However, Multics performance in this area has been less than optimal and there clearly exists a need to improve the situation. Even within Project MAC there are users who regularly ship files back and forth between Multics and other Project MAC computer systems. Recently the Cambridge Information Systems Laboratory (CISL) development Multics system was connected to the ARPA Network. This has generated considerable interest among the members of the CSR (Computer Systems Research) and CISL groups in using the Multics file transfer facility for transmission of Multics system tapes (MST) from one Multics system to the other. At currently realizable data transfer rates, transmission of a typical MST, (approximately 12 million bits), takes about half an hour. The usefullness of the present file transfer system is severely limited by its low bandwidth.

One of the reasons for such a poor performance is the output buffering strategy in the Multics IMP DIM (IMP Device Interface Manager.) With the hope of making a significant performance improvement, we recently changed the IMP DIM buffering strategy. To assess the effects of this change an experiment was conducted between the service machine (MIT Multics) and the development machine (CISL Multics.) This memo reports the experiment and its results.

PROBLEM

Due to reasons that are of historical interest only, the output buffers in the IMP DIM have been very small; each buffer can accomodate up to 940 bits only. With the addition of a 72 bit long header, the resulting message has a maximum length of 1012 bits, which in the Network terminology is a single packet message. Due to this buffer size limit, Multics can transmit single packet messages only, even though the network can accept up to 8096 bit long messages. This results in increased overhead in the set up time for transmission of large number of bits.

An old version of the IMP-to-Host protocol requires that a host may not transmit another message on a network connection unless a Request-for-Next-Message (RFNM) has been received in response to the previous message. Even though this restriction has now been relaxed, the protocol does not specify any way to recover from transmission errors that occur while more than one RFNM is pending on the same connection. If the constraint of transmitting only one message at a time is observed there is at least some potential for recovery in case of an error. Being reliability concious, Multics observes the constraint imposed by the old protocol. Following the old protocol introduces delays in the transmission of successive messages on the same link and therefore the overall

bandwidth per connection is reduced.

SOLUTION

At least one method of improving the file transfer bandwidth is obvious. Increasing the size of IMP DIM output buffers will allow us to transmit significantly larger messages.[1] This will result in fewer RFNMs and delays and improved bandwidth. We have made two assumptions here. The first assumption is that the delay introduced by RFNMs does not increase with the size of the message.[2] Our experience with the network tends to support this assumption. The second assumption is that actual time taken to transmit a message increases, at best, linearly with the size of message. This second assumption does not hold for a heavily loaded network. But for our purpose we may assume it to be essentially correct.

There is another advantage in transmitting large messages. For a given amount of data, fewer messages will be transmitted, fewer RFNMs will be read, and therefore time spent in the channel driver programs will be reduced. Since the channel sits idle during at least some of this time, the idle time for the channel will be reduced, resulting in improved channel bandwidth.

EXPERIMENT

We changed the IMP DIM to implement two kinds of output buffers. One kind of output buffer is short and can hold single packet messages only. The other kind of buffer is, naturally enough, large and can hold the largest message allowed by the network. If the number of bits to be transmitted is low (this is mostly the situation with interactive users,) a small buffer is chosen and if it is large (for example, file transfers,) a large buffer is chosen.

The new IMP DIM was used in an experimental Multics system on the development machine at CISL. The service machine at MIT had the old version. Large files were transmitted from the development system to the service system and the file transfer rate was measured in bits per second (of real time.) To get an estimate of gain in the performance, files were transmitted from the service

---

[1] In one situation it may not be possible to transmit large messages. The number of messages and bits that a host can transmit on a particular connection must not exceed the message and bit allocation provided by the receiving host. If a receiving host gives very small bit allocation, then the sending host can not transmit very large messages. Since Multics always gives out very large allocations, there should be no problem in Multics to Multics file transfers.

[2] It should be noted that the size of RFNM is fixed and does not change with the size of message.

system to the development system and the old file transfer rate was measured. The same file, approximately 2.5 million bits long, was used in both experiments. The BYTE-SIZE and MODE parameters of the ARPANET File Transfer protocol were set to 36 and "image" mode respectively. This implies that no character conversion was performed in the file transfer. The following table shows the results obtained:

	Service to Development (old system)	Development to Service (new system)
average file-transfer rate (bits per second)	6,837	27,578

The following information regarding the environment of the experiment is supplied for the sake of completeness. At the time of this experiment, the service system was lightly loaded (the system load varied between 30.0 and 35.0). The service system configuration was 2 cpu's and 384 K primary memory. The development machine configuration was 1 cpu and 256 K of primary memory. The development system load was limited to the file transfer processes and the initializer process. The MIT Multics is connected to the IMP number 44 (port 0) by a rather short cable (approximately 100 feet long.) The CISL Multics is connected to the IMP number 6 (port 0) by an approximately 1500 feet long cable. Both IMPs are in close physical proximity (approximately 2000 feet,) and are connected to each other by a 50 kilobits per second line. The results given above show considerable improvement in the performance with the new IMP DIM.

Since there is considerable interest in the Multics development community in using the file transfer facility for transmitting Multics System Tapes between the two systems, we are providing here an estimate of time that would be taken to transmit the current MST. MST 23.4-0a contains 334,231 words. When multiplied by 36 (the word length in bits) this yields the total number of bits to be approximately 12.5 million. Assuming a file transfer rate of 27,500 bits per second, it will take approximately 7 minutes and 30 seconds to transmit the system 23.4-0a.

In the experiment outlined above, there was only one file being transferred at any given time between the two systems. We conducted another experiment to get an estimate of performance in the situation of multiple file transfers occurring simultaneously. This experiment consisted of first two, and then three, simultaneous file-transfers from the development system to the service system. These file transfers were started by different processes logged in from separate consoles. Because these file-transfers were started manually, we could not obtain perfect simultaneity and results obtained for the total bandwidth are essentially approximate. For two simultaneous file transfers the total bit rate was approximately 40,000 bits per second and bit rate for each file transfer was approximately 20,000 bits per second. For three simultaneous file transfers, total bit rate remained at approximately 40,000 bits per second and bit rate for individual transfers was approximately 13,500 bits per second.